Connectivity-based error evaluation for ellipse fitting

Tomonari Masuzaki, Yasuyuki Sugaya
Department of Computer Science and Engineering, Toyohashi University of Technology
Toyohashi, Aichi 441-8580 Japan
{masuzaki, sugaya}@iim.cs.tut.ac.jp

Abstract

We propose a new method for fitting an ellipse to a point sequence extracted from an image. This method can fit an ellipse if a point sequence consists of elliptic arcs and non-elliptic arcs such as line segments. Assuming that input points are spatially connected, we iteratively select inlier points and fit an ellipse to them by computing curvatures of the residual graph. By using simulated data and real images, we compare the performance of our method with existing methods and show that the accuracy and computation time of the proposed method is superior to existing methods.

1 Introduction

A circular object in a scene is projected onto the image plane as an ellipse, and we can compute its 3-D positions from that ellipse [4]. Therefore, detecting circles and ellipses in images is the first step of many computer vision applications including industrial robotic operations and autonomous navigation. For this purpose, many methods for extracting elliptic arcs from an image and for fitting an ellipse to the extracted elliptic arcs are studied [5, 12, 14].

Many methods for fitting an ellipse to the obtained elliptic arcs have been proposed in the past [8, 9, 1, 10, 6, 3, 13, 11]. However, most of them do not consider the presence of non-elliptic arcs, which we call *outliers*, in the input data.

Methods to deal with outliers are classified into two approaches. The one approach removes outlier data from the input data before applying an ellipse fitting method. The other approach detects outliers in the course of ellipse fitting. In the former approach, line fitting based methods and curvature-based methods exist. However, these methods cannot always remove all outliers. In the latter approach, RANSAC is a well known framework for dealing with outliers [2]. Yu et al. [14] detected an outlier point sequence from fitting errors and removed it from the input data. By iteratively applying the above procedure, they fitted an ellipse to the remaining inliers.

In this paper, we propose a new method for simultaneously fitting an ellipse and detecting outliers. Assuming that input data are a spatially connected sequence of edge points, we segment it into partial arcs by considering the ellipse fitting residuals and detect inliers by computing the curvature of the residual graph of each of the segmented arcs.

Our method has several advantages over existing methods. Our method involves iterations, but the number of iterations is much less than those of RANSAC and Yu's method. Moreover, in contrast to Yu's method, our method has the possibility of fitting a more accurate ellipse because outliers are removed from the input data in each iteration step, meaning that the number of data to fit an ellipse does not decrease by iterations.

2 Ellipse fitting

Curves represented by a quadratic equations in x and y in the form

$$Ax^2 + 2Bxy + Cy^2 + 2f_0(Dx + Ey) + f_0^2F = 0, \quad (1)$$

are called *conics*, which include ellipses, parabolas, hyperbolas, and their degeneracies such as two lines [4].

Our task is to compute the coefficients $A, ..., F$ so that the ellipse of Eq. (1) passes through the detected points $(x_\alpha, y_\alpha), \alpha = 1, ..., N$, as closely as possible. In Eq. (1), f_0 is a constant that has the order of the image size for stabilizing finite length numerical computation[1]. For a point sequence (x_α, y_α), $\alpha = 1, ..., N$, we define 6-D vectors

$$\begin{aligned} \boldsymbol{\xi}_\alpha &= (x_\alpha^2,\, 2x_\alpha y_\alpha,\, y_\alpha^2,\, 2f_0x_\alpha,\, 2f_0y_\alpha,\, f_0^2)^\top, \\ \boldsymbol{\theta} &= (A,\, B,\, C,\, D,\, E,\, F)^\top. \end{aligned} \quad (2)$$

The condition that (x_α, y_α) satisfies Eq. (1) is written as

$$(\boldsymbol{\xi}_\alpha, \boldsymbol{\theta}) = 0, \quad (3)$$

where $(\boldsymbol{a}, \boldsymbol{b})$ denotes the inner product of vectors $\boldsymbol{a}$ and $\boldsymbol{b}$. Since vector $\boldsymbol{\theta}$ has scale indeterminacy, we normalize it to unit norm: $\|\boldsymbol{\theta}\| = 1$.

Since Eq. (3) is not exactly satisfied in the presence of noise, we compute a $\boldsymbol{\theta}$ such that $(\boldsymbol{\xi}_\alpha, \boldsymbol{\theta}) \approx 0$, $\alpha = 1, ..., N$. For computing a $\boldsymbol{\theta}$ that is close to its true value, we need to consider the statistical properties of noise. The standard model is to regard the noise in (x_α, y_α) as an independent Gaussian random variable of mean 0 and standard deviation σ. Then, the covariance matrix of the vector $\boldsymbol{\xi}_\alpha$ has the form $\sigma^2 V_0[\boldsymbol{\xi}_\alpha]$, where

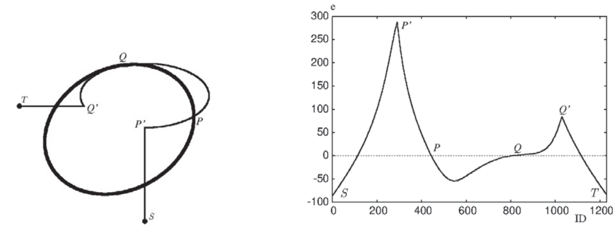
$$V_0[\boldsymbol{\xi}_\alpha] = 4\begin{pmatrix} x_\alpha^2 & x_\alpha y_\alpha & 0 & f_0x_\alpha & 0 & 0 \\ x_\alpha y_\alpha & x_\alpha^2 + y_\alpha^2 & x_\alpha y_\alpha & f_0y_\alpha & f_0x_\alpha & 0 \\ 0 & x_\alpha y_\alpha & y_\alpha^2 & 0 & f_0y_\alpha & 0 \\ f_0x_\alpha & f_0y_\alpha & 0 & f_0^2 & 0 & 0 \\ 0 & f_0x_\alpha & f_0y_\alpha & 0 & f_0^2 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix}, \quad (4)$$

which we call the *normalized covariance matrix* [6].

3 Proposed method

We assume that input data are a spatially connected curve and that fitted ellipse intersects the curve at multiple points (Fig. 1(a)). We segment the input curve

[1] We set $f_0 = 600$.



(a) (b)

Figure 1. (a) An ellipse fitted to a point sequence which includes non-elliptic arcs. A thin curve is an input point sequence. (b) Fitting residual graph. The horizontal axis shows the index of the points, which starts from the point S to the point T shown in (a). The vertical axis shows the signed fitting residual whose sign is computed by the left-hand side of Eq. (3).

into partial arcs at these intersection points. We compute the variation of the tangent angle to the graph of the fitting residual, which we simply call *error curvature* in this paper, and judge if each arc is an inlier or an outlier based on this *error curvature*. The principle of our method is nearly equivalent to the curvature-based outlier detection. However, our method is more efficient, because our method computes the curvature of the residual graph at only one point where the residual takes a maximum in each segmented arc.

In Fig. 1, the input point sequence is divided into five partial arcs by the fitted ellipse. The residual value of the arc PQ, which consists only of an elliptic arc, smoothly changes around the peak value. On the other hand, we can see that the residual graph has a peaky shape over the arcs consisting of non-elliptic arcs.

Moreover, we can see that the arcs PP' and QQ', which are connected to the elliptic arc PQ, are also elliptic arcs. The points P' and Q' are the peak points of the partial arcs adjacent to the elliptic arc PQ. Therefore, if we use not only the detected inlier arc but also the adjacent arcs like the arcs PP' and QQ' for ellipse fitting, we can effectively fit a correct ellipse. The algorithm of our method is summarized as follows:

1. Fit an ellipse to a point sequence by Fitzgibbon's method [3].
2. Compute the sign of left-hand side of Eq. (3) for all the points and segment the point sequence into partial arcs at the points across which the computed sign changes.
3. For each segmented arc, detect the point where the residual value takes a maximum and compute its error curvature ϕ at this point.
4. Go to Step (a) if it is the first ellipse fitting, else go to Step (b).
   (a) Select an inlier arc which has the smallest value ϕ among those arcs whose arc lengths are longer than a threshold[2] and adjacent arcs if their end points correspond to the peak of the arc. Then, we fit an ellipse to these selected arcs.
   (b) Select the arcs whose error curvature ϕs are smaller than a threshold $\hat{\phi}$[3] and fit an ellipse to them.
5. Repeat the procedures from Step 2 to Step 4 until the number of inliers does not change.

As discussed before, if the selected arc is an elliptic arc, the adjacent arcs are also elliptic arcs. Therefore, we can effectively fit a correct ellipse if we use those arcs. However, if we select a non-elliptic arc as an inlier and add adjacent outlier arcs to fit an ellipse, we cannot fit a correct ellipse.

For this reason, in the first iteration of our algorithm, we select among the arcs that are sufficiently long the one whose error curvature ϕ at its peak point is the smallest. We regard it as a reliable inlier and extend it to the adjacent partial arcs. After the first iteration, we select all arcs whose error curvature ϕs are smaller than a threshold $\hat{\phi}$. We do not extend those arcs, because the adjacent arcs are non-elliptic arcs if the selected inlier arcs approximately belong to the correct ellipse.

In the following sections, we describe details of our method.

4 Division of the point sequence

The left-hand side of Eq. (3) at the point p_α has a different sign outside and inside the ellipse θ. Using this fact, we can segment the input point sequence $\{p_\alpha\}$ into partial arcs at those point p_α which have different signs from their neighboring points.

Numerically, however, the value $(\boldsymbol{\xi}_\alpha, \boldsymbol{\theta})$ may not be exactly zero even if the point p_α lies on the fitted ellipse θ; the points lying on the fitted ellipse may irregularly change their signs, resulting, so these points may be divided in very short arcs. To avoid this, we regard those partial arcs whose fitting error is close to zero as elliptic arcs, and judge that they are inlier arcs without computing its curvature. We call such arcs *tangent arcs*.

5 Inlier arc selection

After segmenting the point sequence, we detect the point where the fitting error takes its maximum, and compute its *error curvature* at this point in each segmented arc. If the computed curvature is larger than a threshold, we regard this arc as an outlier. The numerical value of the curvature of the residual graph depends on the scale of the horizontal axis of the residual graph. For example, if two point sequences have the same shape and different scales, the curvatures of their sequences have different values according to our definition. Therefore, we normalize the scale of the horizontal axis of the residual graph in the form

$$\boldsymbol{Q}_\alpha = (\frac{\lambda e_{max} \alpha}{N}, e_\alpha)^\top, \qquad (5)$$

where e_{max} is the maximum of all ellipse fitting residuals and λ is a constant for normalization[4]. The fitting

[2] We set the threshold to be 5% of the number of input edge points.

[3] We set $\hat{\phi} = 80°$.

[4] We set λ=2.0.

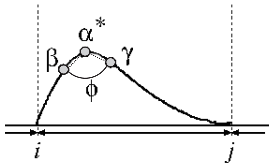


Figure 2. Error curvature ϕ of the peak point α^* for a partial arc.

residual e_α at a point p_α is computed by

$$e_\alpha = \sqrt{\frac{(\boldsymbol{\xi}_\alpha, \boldsymbol{\theta})}{(\boldsymbol{\theta}, V_0[\boldsymbol{\xi}_\alpha]\boldsymbol{\theta})}}. \quad (6)$$

If e_{max} is extremely large, the normalization of Eq. (5) may not work well. So, if e_{max} is larger than a threshold E_{max}[5], we replacing the value e_{max} with the threshold E_{max} in the normalization computation.

For M segmented arcs $\boldsymbol{R}_\kappa(i,j) = \{\boldsymbol{Q}_\beta | \beta = i, ..., j\}$,$\kappa = 1, ..., M$, we select inlier arcs by the following algorithm. Figure 2 summarizes the symbols used in the following algorithm.

1. Let α^* be the index of the point whose residual takes its maximum in the arc $\boldsymbol{R}_\kappa$. Here, if the residual at the point p_{α^*} is smaller than a threshold E_{min}[6], we regard the arc $\boldsymbol{R}_\kappa$ as a *tangent arc* and finish this procedure.

2. Select two points whose indices β and γ are such that

$$\beta = \alpha^* - d, \ \gamma = \alpha^* + d, \ d = (j-i)/r, \quad (7)$$

   where r[7] is a constant for determining the distance between the point α^* and its adjacent points β and γ. If both β and γ are out of $\boldsymbol{R}_\kappa$, we update d to $d \leftarrow d-1$ until either of the two points are in the arc $\boldsymbol{R}_\kappa$.

3. Compute two vectors $\boldsymbol{x}^{(1)}$ and $\boldsymbol{x}^{(2)}$ according to the following three rules. Here, $x_b^{(a)}$ denotes the b-th component of the vector $\boldsymbol{x}^{(a)}$.

   case a: Points $\boldsymbol{Q}_\beta$ and $\boldsymbol{Q}_\gamma$ are both in the arc $\boldsymbol{R}_\kappa$.

$$\boldsymbol{x}^{(1)} = \boldsymbol{Q}_{\alpha^*} - \boldsymbol{Q}_\beta, \ \boldsymbol{x}^{(2)} = \boldsymbol{Q}_{\alpha^*} - \boldsymbol{Q}_\gamma. \quad (8)$$

   case b: Point $\boldsymbol{Q}_\beta$ is in the arc $\boldsymbol{R}_\kappa$.

$$\boldsymbol{x}^{(1)} = \boldsymbol{Q}_{\alpha^*} - \boldsymbol{Q}_\beta, \ \boldsymbol{x}^{(2)} = (-x_1^{(1)}, x_2^{(1)})^\top. \quad (9)$$

   case c: Point $\boldsymbol{Q}_\gamma$ is in the arc $\boldsymbol{R}_\kappa$.

$$\boldsymbol{x}^{(2)} = \boldsymbol{Q}_{\alpha^*} - \boldsymbol{Q}_\gamma, \ \boldsymbol{x}^{(1)} = (-x_1^{(2)}, x_2^{(2)})^\top. \quad (10)$$

4. Compute the error curvature ϕ by

$$\phi = \pi - \cos^{-1}\left(\frac{(\boldsymbol{x}^{(1)}, \boldsymbol{x}^{(2)})}{||\boldsymbol{x}^{(1)}|| ||\boldsymbol{x}^{(2)}||}\right). \quad (11)$$

5. Regard the arc $\boldsymbol{R}_\kappa$ as an inlier arc if ϕ is smaller than the threshold $\hat{\phi}$.

We extend the selected inlier arc to its adjacent arcs to generate a longer arc in the first round of ellipse fitting. If the adjacent arc is a *tangent arc*, we test the next adjacent arc and regard the non-*tangent arc* as an inlier arc.

6 Experiment

6.1 Simulations

In order to confirm the effectiveness of the our proposed method, we compared our method with RANSAC and Yu's method.

Figure 3 shows the experimental results. The thick curve of Fig. 3(a) is the ellipse fitted by our method and the thick line of Fig. 3(b) shows the selected inlier points for fitting the final ellipse (Fig. 3(a)). Figure 3(c), (d) and (e), (f) show the results of RANSAC and Yu's method, respectively, in the same manner as Fig. 3(a) and (b). We applied three methods to different variations of input data and Fig. 3(1), (2), and (3) are three examples of them.

As we can see that our method could fit correct ellipses even if outliers are smoothly connected an elliptic arc point sequence shown in Fig. 3(1). RANSAC also fitted correct ellipses for Fig. 3(1) and (3). However, degenerated conic, which is two lines, is fitted for Fig. 3(2). Yu's method fitted a slightly deviated ellipse from the correct one for Fig. 3(2) and a small ellipse to a short arc for Fig. 3(3).

Table 1 shows the number of iterations and computation times for three methods. We used Intel Core 2Duo 3.00GHz×2 for the CPU with main memory 4GB and Ubuntu 12.04 for the OS. For RANSAC, we stopped if the solution did not change after 50 consecutive iterations and counted the mean total number of iterations over 10 trials. From this result, the number of iterations and the computation time of our method is superior to RANSAC and Yu's method.

6.2 Real image experiment

Figure 4(a) is an input image and Fig. 4(b) shows the extracted edge points by canny operator. We removed successive edge points whose lengths were shorter than 50 pixels. We selected edge points shown by thick points in Fig. 4(b) and fitted an ellipse to them by three methods, respectively. Figure 4(c), (d), and (e) are the fitted ellipses by our method, RANSAC, and Yu's method, respectively. As we can see that RANSAC and Yu's method did not fit a correct ellipse. On the other hand, the proposed method fitted a correct ellipse.

[5] We set E_{max} be (maximum image coordinates of the input points)/3.

[6] We set $E_{min} = 1.5$

[7] We set $r = 8$.

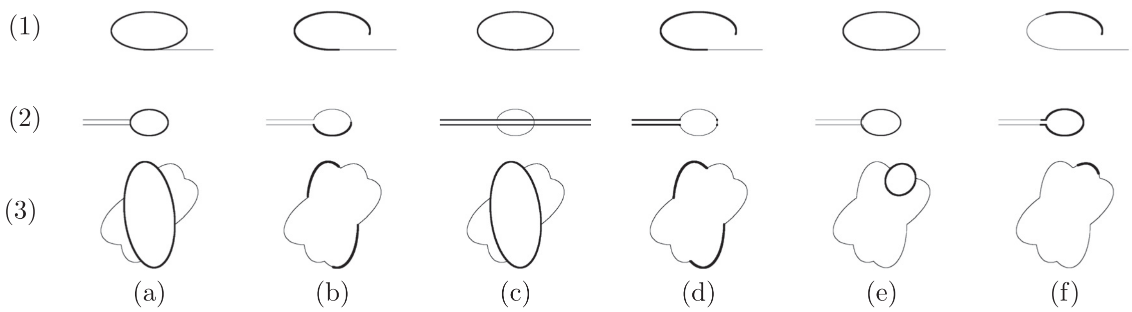


Figure 3. Ellipse fitting results. (a), (c) and (e) The thick curves are the fitted ellipses by our method, RANSAC, and Yu's method, respectively. (b) and (f) The thick points are the used points to fit those ellipse. (d) The thick points are the detected inliers by RANSAC.

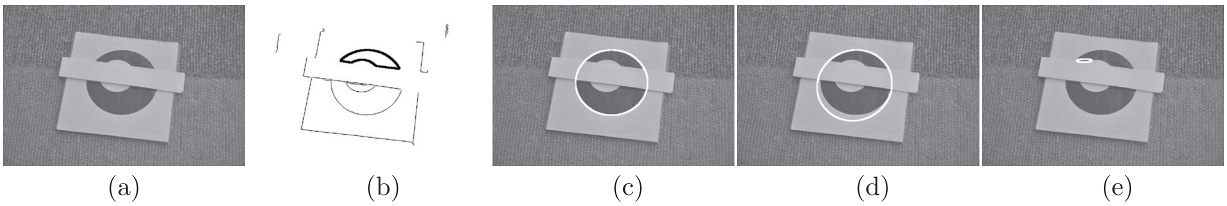


Figure 4. Real image experiment. (a) Input image. (b) Detected edge segments. (c) Our method. (d) RANSAC. (e) Yu's method.

Table 1. Comparison of computation time and number of iterations: Computation time in msec (number of iterations).

	Our method	RANSAC	Yu's method
(1)	8 (9)	42 (148)	600 (178)
(2)	4 (3)	64 (143)	424 (157)
(3)	8 (8)	68 (134)	48 (10)

7 Concluding remarks

We proposed a new method for fitting ellipse to a point sequence which contains non-elliptic arcs. Assuming that input points are a spatially connected sequence of edge points, we segment it to partial arcs by considering the ellipse fitting residuals and detect inlier arcs by computing the curvature of the residual graph of each segmented arc.

By using simulated data and real images, we compared the performance of our method with RANSAC and Yu's method and showed that the accuracy and computation time of the proposed method were superior to existing methods.

Acknowledgements. The authors thank Kenich Kanatani of Okayama University, Japan for helpful discussions. This work was supported in part by JSPS Grant-in-Aid for Scientific Research(C) (26330192).

References

[1] W. Chojnacki, M. J. Brooks, A. van den Hengel and D. Gawley, On the fitting of surfaces to data with covariances, IEEE Trans. Patt. Anal. Mach. Intell., **22**-11, pp. 1294–1303, 2000.

[2] M. A. Fischler and R. C. Bolles, Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography, Comm. ACM, **24**-6, pp. 381–395, 1981.

[3] A. Fitzgibbon, M. Pilu, and R. B. Fisher, Direct least squares fitting of ellipses, IEEE Trans. Patt. Anal. Mach. Intell., **21**-5 pp. 476–480, 1999.

[4] K. Kanatani, Geometric Computation for Machine Vision, Oxford University Press, Oxford, U.K., 1993.

[5] K. Kanatani and N. Ohta, Automatic detection of circular objects by ellipse growing, *Int. J. Image Graphics*, Vol. 4, No. 1, pp. 35–50, 2004.

[6] K. Kanatani and Y. Sugaya, Performance evaluation of iterative geometric fitting algorithms, Comput. Stat. Data Anal., **52**-2, pp. 1208–1222, 2007.

[7] K. Kanatani and Y. Sugaya, Unified computation of strict maximum likelihood for geometric fitting, J. Math. Imaging Vis., **38**-1, pp. 1–13, 2010.

[8] K. Kanatani, P. Rangarajan, Y. Sugaya and H. Niitsuma, HyperLS and its applications, IPSJ Trans. Comput. Vis. Appl., **3**, pp. 80–94, 2011.

[9] K. Kanatani, A. Al-Sharadqah, N. Chernov, and Y. Sugaya, Renormalization Returns: Hyper-renormalization and Its Applications, Proc. 12th Euro. Conf. Comput. Vis., **3**, pp. 385–398, 2012.

[10] Y. Leedan and P. Meer, Heteroscedastic regression in computer vision: Problems with bilinear constraint, Int. J. Comput. Vision, **37**-2, p. 127–150, 2000.

[11] T. Masuzaki, Y. Sugaya and K. Kanatani, High accuracy ellipse-specific fitting, 6th Pacific-Rim Symposium on Image and Video Technology (PSIVT 2013), pp. 314–324, 2013.

[12] Y. Sugaya, Ellipse Detection by combining division and model selection based integration of edge points, 4th Pacific-Rim Symposium on Image and Video Technology(PSIVT), pp. 64–69, 2010.

[13] Z. L. Szpak, W. Chojnacki, and A. van den Hengel, Guaranteed ellipse fitting with Sampson distance, Proc. 12th Euro. Conf. Comput. Vis., **5**, pp. 87–100. 2012.

[14] Yu Qiao, and S. H. Ong, Arc-based evaluation and detection of ellipses, Journal of Pattern recognition, **40**-7, pp. 1990–2003, 2007.